Cover Page

Universiteit Leiden

The following handle holds various files of this Leiden University dissertation:
http://hdl.handle.net/1887/81487

Author: Mechev, A.P.
Title: Orchestration of Distributed LOFAR Workflows
Issue Date: 2019-12-09

Orchestration of Distributed LOFAR Workflows

PROEFSCHRIFT

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden
op gezag van de Rector Magnificus Prof. mr. C.J.J.M Stolker,
volgens besluit van het College voor Promoties
te verdedigen op maandag 9 December 2019 klokke 15:00 uur

door
Alexandar Plamenov Mechev
geboren te Sofia, Bulgarije
in 1989

Promotiecommissie:

Promotor:	Dhr. Prof.dr. H.J.A. Röttgering
Promotor:	Dhr. Prof.dr. A. Plaat
Co-Promotor:	Dr. J.B.R. Oonk
Overige leden:	Dhr. Prof.dr. S.F. Portegies Zwart
	Dhr. Prof.dr. H.A.G. Wijshoff
	Prof.dr. Rob van Nieuwpoort
	Prof.dr. Martin Hardcastle
	Dr. A.L. Varbanescu
	Dhr. Dr.ing. H.T. Intema
	Dhr. Dr. T.W. Shimwell

The cover of this thesis depicts a parent duck orchestrating a neat line of ducklings, inspired by a common sight during the author's commute by bike. The theme of this work, orchestration of complex workflows, can be expressed by the idiom "having your ducks in a row", and the cover page illustrates this.

Contents

1 Introduction **1**

1.1 Introduction 1

1.2 LOFAR 6

1.3 Problem Statement and Research Questions 12

1.4 Contributions 14

2 LOFAR-DSP platform **17**

2.1 Introduction 17

2.2 LOFAR observations 19

2.3 LOFAR Processing 21

2.4 LOFAR-DSP 23

2.5 Executing LOFAR-DSP 29

2.6 Deploying LOFAR-DSP 32

2.7 Discussion 35

2.8 Conclusions 38

3 LOFAR Scalability Framework **43**

3.1 Introduction 44

3.2 Related Work 45

3.3 LOFAR Data Processing 46

3.4 Framework Design 49

3.5 Conclusion and Future Work 54

3.A Execution of LOFAR Reduction Tools 57

4 Pipeline Collector **59**

4.1 Introduction 60

4.2 Measuring LOFAR Pipeline performance with pipeline_collector 63

4.3 LOFAR Prefactor Test Case 66

4.4 CPU Utilization Tests with PAPI 73

4.5 Discussions and Recommendations 77

4.6 Conclusions 79

4.A Performance Collection Implementation Details 80

5 Fast and Reproducible LOFAR Workflows with AGLOW **83**

5.1 Introduction 84

5.2 Background 85

5.3 Related Work 86

5.4 AGLOW 87

5.5 Results and Discussions 96

5.6 Conclusions 97

6 Scalability Model for the LOFAR Direction Independent Pipeline **101**

6.1 Introduction 102

6.2 Related Work 104

6.3 Processing Setup 104

6.4 Results 109

6.5 Discussions and Conclusions 122

6.6 Applications and Conclusions 127

6.A Calibration Solutions for the sky model tests 128

6.B Parametric model parameters and fit accuracy 129

7 Automated testing and quality control of LOFAR scienctific pipelines with AGLOW **133**

7.1 Introduction 133

7.2 Background 136

7.3 Related Work 137

7.4 Automated testing with AGLOW 138

7.5 Results 141

7.6 Discussion and Conclusions 145

8 Conclusion **149**

8.1 Summary of Thesis Contributions 149

8.2 Answers to Research Questions 150

8.3 Limitations 151

8.4 Future Work 152

Acknowledgements **155**

Samenvatting **159**

English Summary **165**

List of Publications **169**

8.5 Journal Articles 169

8.6 Conference Proceedings 170

Bibliography **172**

List of Tables

2.1 Comparison of LOFAR software distribution methods. 27

4.1 A table of all the results presented in Section 4.3. 62
4.2 Hardware specifications of the four test machines. 66

6.1 Averaging parameters and final data sizes for a sample LOFAR Observation 106
6.2 List of test sky models 107
6.3 Image statistics for four different sky models 115
6.4 Queueing statistics per requested number of CPUs 118
6.5 Fit parameters for the models in Equation 6.1. 131
6.6 Goodness of fit parameters for the model in Equation 6.4. 132

List of Figures

1.1 Two supercomputers sixty years apart. 2

1.2 Graphical representation of aperture synthesis 5

1.3 Image of the raw data 10

1.4 Image of preprocessed data 10

1.5 Image of DI calibrated data 11

1.6 Fully calibrated image 11

2.1 Structure of the LOFAR-DSP platform. 40

2.2 Implementation of Pilot Job Launching 41

3.1 Prefactor and DDFacet Data Flow 46

3.2 Parallelization of prefactor processing 49

3.3 Overview of the design of the LRT framework. 50

3.4 Starting processing on worker machines 52

3.5 Schematic of data movement. 53

4.1 The four processing stages that make up the prefactor pipeline. 64

4.2 Portion of processing time taken by each step for the four prefactor stages. . 65

4.3 Job completion times for two processing steps tested on four hardware setups 68

4.4 Speed comparison between natively and remotely compiled software for the 'calib_cal' step. 69

4.5 Speed comparison between natively and remotely compiled software for the 'gsmcal_solve' step. 69

4.6 A model of the memory hierarchy, as described in [104]. 70

4.7 Effect of CPU speeds on the bottle neck steps for the four test machines. . . 71

4.8 Effect of cache size on the bottle neck steps for the four test machines. . . . 72

4.9 Effect of RAM throughput on the bottle neck steps for the four test machines. 73

4.10 Time series of the Virtual Memory Resident Set Size 74

4.11 Performance of the two bottleneck steps and Disk bandwidth in MB/s. . . . 74

4.12 Cache miss rates for the bottleneck steps, executed on the SURFsara gina cluster. 75

4.13 Resource stall cycles and Full Instruction Issue cycles. 76

4.14 Communication between worker nodes and the TSDB server, including the pipeline_collector modules (in red). 81

5.1 Design of the AGLOW software, including its constituent packages. 89

5.2 Graphical representation of the Airflow Operators built for AGLOW . . . 92

5.3 Renderng of the DAG encoding a DPPP parset as shown by the Airflow User Interface. 93

5.4 Render of the DAG encoding the full prefactor pipeline. 98

6.1 The major steps of the `prefactor` DI pipeline. 106

6.2 The size of the sky model (measured as the number of sources) increases exponentially as we decrease the flux cutoff of the model (i.e. increase the sensitivity). 108

6.3 Plots of the run time as a function of input data size 111

6.4 Tests of the `gsmcal_apply` step for data from 1GB to 64GB 112

6.5 Processing time of `gsmcal_solve` vs number of sources in skymodel . . 113

6.6 Run time vs cutoff sensitivity 114

6.7 Comparison of data calibrated with four sky models 114

6.8 The processing time of the `gsmcal_solve` step vs number of CPUs requested 116

6.9 No speed-up for `gsmcal_apply` seen 116
6.10 Queueing tests on the `GINA` cluster 117
6.11 Model of queueing times 119
6.12 Histogram of download times 119
6.13 Histogram of extraction times 120
6.14 Exponential model of data transfer/extraction on the `GINA` cluster 120
6.15 Tet of download/extract times for ten 1GB data sets 121
6.16 Tet of download/extract times for a 64GB data set 122
6.17 Processing time for the `gsmcal_solve` step in a production environment 123
6.18 Comparison of scalability model with production runs 124
6.19 Calibration Solutions for sky models with low flux cutoff 129
6.20 Calibration solution differences between two skymodels 130
6.21 Calibration solutions for sky models with high flux cutoff 130

7.1 Diagram of the `prefactor` Continuous Integration workflow 140
7.2 A test image created on 2019-04-23 by our automated CI workflow showing a diffuse radio source 142
7.3 Diagram of integration of three scientific pipelines with test data and processing infrastructure 142
7.4 Images created by the CI runs from 2019-03-29 and 2019-04-23 showing a bright source 143
7.5 Time series of measurements done with images automatically produced by our workflow 143